

00862.023668.



IFW

PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: ) Examiner: Not Yet Assigned  
TOMOTOSHI KANATSU )  
Application No.: 10/828,470 ) Group Art Unit: Not Yet Assigned  
Filed: April 21, 2004 )  
For: IMAGE PROCESSING METHOD ) June 30, 2004  
AND SYSTEM )

Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

SUBMISSION OF PRIORITY DOCUMENT

Sir:

In support of Applicant's claim for priority under 35 U.S.C. § 119, enclosed  
is a certified copy of the following foreign application:

2003-121076 filed April 25, 2003.

Applicant's undersigned attorney may be reached in our New York office by  
telephone at (212) 218-2100. All correspondence should continue to be directed to our address  
given below.

Respectfully submitted,

  
\_\_\_\_\_  
Attorney for Applicant  
Carl B. Wischhusen

Registration No. 43,279

FITZPATRICK, CELLA, HARPER & SCINTO  
30 Rockefeller Plaza  
New York, New York 10112-3801  
Facsimile: (212) 218-2200

CFM03668  
10/18/2014 470US

日本国特許庁  
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日      2003年  4月25日  
Date of Application:

出願番号      特願2003-121076  
Application Number:

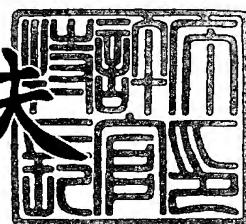
[ST. 10/C] :      [JP2003-121076]

出願人      キヤノン株式会社  
Applicant(s):

2004年 5月14日

特許庁長官  
Commissioner,  
Japan Patent Office

今井康夫



出証番号 出証特2004-3040455

【書類名】 特許願  
【整理番号】 254075  
【提出日】 平成15年 4月25日  
【あて先】 特許庁長官殿  
【国際特許分類】 G06F 12/00  
G06K 9/60  
【発明の名称】 画像処理方法  
【請求項の数】 1  
【発明者】  
【住所又は居所】 東京都大田区下丸子3丁目30番2号キヤノン株式会社  
内  
【氏名】 金津 知俊  
【特許出願人】  
【識別番号】 000001007  
【氏名又は名称】 キヤノン株式会社  
【代理人】  
【識別番号】 100079832  
【弁理士】  
【氏名又は名称】 山本 誠  
【選任した代理人】  
【識別番号】 100078846  
【弁理士】  
【氏名又は名称】 大音 康毅  
【選任した代理人】  
【識別番号】 100087583  
【弁理士】  
【氏名又は名称】 田中 増顕

**【手数料の表示】**

【予納台帳番号】 085177

【納付金額】 21,000円

**【提出物件の目録】**

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 0206918

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 画像処理方法

【特許請求の範囲】

【請求項 1】 入力画像に対応するオリジナルデータファイルを検索するための画像処理方法であって、

ユーザにより入力された情報に基づいて前記入力画像に関する第1の検索情報を取得する第1検索情報取得ステップと、

前記入力画像に含まれる特徴データを第2の検索情報として取得する第2検索情報取得ステップと、

前記第1の検索情報と前記第2の検索情報を用いて、前記入力画像に対応するオリジナルデータファイルを検索する検索ステップと、  
を有することを特徴とする画像処理方法。

【発明の詳細な説明】

【0001】

【発明が属する技術分野】

本発明は画像処理方法に関する。

【0002】

【従来の技術】

近年、環境問題が叫ばれる中、オフィスでのペーパーレス化が急速に進んでいる。

【0003】

[第1従来例]

ペーパーレス化の方法には、例えば、バインダ等で蓄積された紙文書をスキャナ等で読み取り、ラスターデータのイメージとして、ポータブルドキュメントフォーマット（PDFフォーマット）等のコンパクトなファイルに変換して、画像記憶手段に格納するものがある（例えば特許文献1）。

【0004】

[第2従来例]

ペーパーレス化の方法の第2は、機能が拡張された記録装置、MFP（マルチ

ファンクション複合機)を用い、画像記憶装置に文字や画像のオリジナルデータファイルを格納しておき、オリジナルデータファイルを印刷して紙文書に記録する際に、オリジナルデータファイルが存在する画像記憶装置内のポインタ情報を、紙文書の表紙や印刷情報中に付加情報として記録するものがある(例えば特許文献2)。これによって、ポインタ情報から直ちにオリジナルデータファイルにアクセスし、オリジナルデータファイルの編集、印刷等の再利用が可能であり、紙文書保有量を削減し得る。

#### 【0005】

##### 【特許文献1】

特開2001-358863号公報

##### 【特許文献2】

特開平10-285378号公報

#### 【0006】

##### 【発明が解決しようとする課題】

第1従来例は、スキャナで読み取った画像をコンパクトな情報量のPDFファイルとして保存可能であるが、プリントされた文書から保存されているファイルを検索することができず、保存されている文書を再利用することは難しかった。

#### 【0007】

第2従来例は、オリジナルデータファイルへのポインタ情報を持たない文書ファイルの場合は、オリジナルデータファイルを検索することができない。

#### 【0008】

本発明はこのような問題点を解消すべく創案されたもので、紙文書をスキャンするなどして得たイメージデータに基づいて、オリジナルデータファイルを検索するものであって、特に、入力画像に対応するオリジナルデータファイルを検索する際に、より高精度に検索することを目的とする。

#### 【0009】

また、オリジナルデータファイルが見つからなかった場合は、該入力画像をベクトルデータ化してデータベースに格納する。

#### 【0010】

**【課題を解決するための手段】****【0011】**

本発明に係る、画像処理方法は、入力画像に対応するオリジナルデータファイルを検索するための画像処理方法であって、ユーザにより入力された情報に基づいて前記入力画像に関する第1の検索情報を取得する第1検索情報取得ステップと、前記入力画像に含まれる特徴データを第2の検索情報として取得する第2検索情報取得ステップと、前記第1の検索情報と前記第2の検索情報とを用いて、前記入力画像に対応するオリジナルデータファイルを検索する検索ステップと、を有する。これによって、紙文書に対応するオリジナルデータファイルを検索する際の検索性能が向上する。

**【0012】****【発明の実施の形態】**

次に本発明に係る画像処理方法の一実施形態を図面に基づいて説明する。

**【0013】**

図1は、本発明に係る画像処理方法の第1実施形態を実施する画像処理システムを示すブロック図、図2は、図1におけるMFPを示すブロック図、図3は、本発明に係る画像処理方法の一実施形態を示すフローチャート、図4は、図3の画像処理方法により処理される原稿と処理結果の画像を示す図、図5は、ブロックセレクション処理で得られるブロック情報および入力ファイル情報を示す表、図6は、原稿の画像からポインタ情報を抽出する処理を示すフローチャート、図7は、ポインタ情報を含む原稿の画像を示す図、図8は、図3のポインタ情報に基づくファイル検索処理を示すフローチャート、図9は、図3の文字領域のベクトル化の処理を示すフローチャート、図10は、図3のファイル検索の処理を示すフローチャート、図11は、図9のベクトル化の処理における角抽出の処理を示す図、図12は、図9のベクトル化の処理における輪郭線まとめの処理を示す図、図13は、図9のベクトル化で生成されたベクトルデータのグループ化の処理を示すフローチャート、図14は、図13の処理でグループ化されたベクトルデータに対する図形要素検出の処理を示すフローチャート、図15は、ベクトル化処理結果のデータを示すマップ、図16は、図3のアプリデータ変換の処理を

示すフローチャート、図17は、図16の文書構造ツリー生成の処理を示すフローチャート、図18は、文書構造ツリー生成処理の対象となる文書を示す図、図19は、図17の処理によって生成される文書構造ツリーを示す図、図20は、図3のポインタ情報付加の処理を示すフローチャート、である。

#### 【0014】

##### [画像処理システム]

図1において、本発明に係る画像処理方法の一実施形態を実施する画像処理システムは、オフィス10とオフィス20とをインターネット104で接続した環境において使用される。

#### 【0015】

オフィス10内に構築されたLAN107には、記録装置としてのMFP（マルチファンクション複合機）100、MFP100を制御するマネージメントPC101、クライアントPC（外部記憶手段）102、文書管理サーバ106、文書管理サーバ106のためのデータベース105が接続されている。

#### 【0016】

オフィス20内にはLAN108が構築され、LAN108には文書管理サーバ106、および文書管理サーバ106のためのデータベース105が接続されている。

#### 【0017】

LAN107、108にはプロキシサーバ103が接続され、LAN107、108はプロキシサーバ103を介してインターネットに接続される。

#### 【0018】

MFP100は紙文書から読み取った入力画像に対する画像処理の一部を担当し、処理結果としてのイメージデータはLAN109を通じてマネージメントPC101に入力される。マネージメントPC101は、画像記憶手段、画像処理手段、表示手段、入力手段等を含む通常のコンピュータであり、機能的にはこれら構成要素の一部がMFP100と一体化して、画像処理システムの構成要素となっている。なお、本実施形態では、マネージメントPCにおいて、下記に記載の検索処理などが実行されるものとするが、マネージメントPCで行われる処理

をMFPで実行するようにしても構わない。

### 【0019】

さらにMFP100は、LAN109によってマネージメントPC101に直接接続されている。

### 【0020】

#### [MFP]

図2において、MFP100は、図示しないオートドキュメントフィーダ（ADF）を有する画像読み取り部110を備え、画像読み取り部110は束状の或いは1枚の原稿の画像を光源で照射し、反射画像をレンズで固体撮像素子上に結像する。固体撮像素子は所定解像度（例えば600dpi）および所定濃度レベル（例えば8ビット）の画像読み取り信号を生成し、画像読み取り信号からラスターデータよりなるイメージデータが構成される。

### 【0021】

MFP100は、記憶装置111および記録装置112を有し、通常の複写機能を実行する際には、イメージデータをデータ処理装置115によって画像処理して記録信号に変換する。複数枚複写の場合には、1頁分の記録信号を一旦記憶装置111に記憶保持した後、記録装置112に順次出力して、複写紙上に記録画像を形成する。

### 【0022】

MFP100は、LAN107との接続のためのネットワークIF114を有し、クライアントPC102が出力したイメージデータを、記録装置112によって記録し得る。クライアントPC102から出力されたイメージデータはLAN107からネットワークIF114を経てデータ処理装置115で記録可能な記録信号に変換された後、MFP100において、記録紙上に記録画像として記録される。

### 【0023】

MFP100は、MFP100に設けられたキー操作部（入力装置113）、あるいはマネージメントPC101の入力装置（キーボード、ポインティングデバイス等）を通じて操作される。これらの操作のために、データ処理装置115

は内部の制御部（図示せず。）によって所定の制御を実行する。

#### 【0024】

MFP100は表示装置116を有し、操作入力の状態と、処理すべきイメージデータとを、表示装置116によって表示し得る。

#### 【0025】

記憶装置111はネットワークIF117を介して、マネージメントPC101から直接制御し得る。LAN109は、MFP100とマネージメントPC101との間のデータの授受、制御信号授受に用いられる。

#### 【0026】

[画像処理方法の全体の流れ]

本発明に係る画像処理方法の実施形態は図3の各ステップにより実行される。

#### 【0027】

ステップS301：MFP100の画像読み取り部110を動作させて、1枚の原稿をラスター状に走査し、所定解像度および所定濃度レベルの、画像読み取り信号を取得する。画像読み取り信号はデータ処理部115によって前処理され、入力画像の1頁分のイメージデータとして、記憶装置111に保存される。次にステップS302に進む。

#### 【0028】

ステップS302：画像読み取りと同時に、操作画面116においてユーザに入力画像特定のための情報入力を促し、ユーザが情報を入力するか否かを判断する。ユーザが情報を入力するときはステップS303に進み、ユーザが情報を入力しないときはステップS304にジャンプする。

#### 【0029】

ステップS303：ユーザがオリジナルデータファイル特定のための情報をマニュアル入力する。入力される情報は、入力画像のキーワード、入力画像に対応するオリジナルデータファイルのデータサイズ、オリジナルデータファイルの作成日付等、その他入力画像検索に有効な情報である。

#### 【0030】

ステップS304（ブロックセレクション（領域分割）ステップ）：マネージ

メントPC101によって、記憶装置111に格納されたイメージデータの領域を、文字あるいは線画を含む文字・線画領域と、ハーフトーンの写真領域、不定形の画像領域その他に分ける。さらに文字・線画領域について、主に文字を含む文字領域と、主に表、図形等を含む線画領域とを分離し、線画領域は表領域と図形領域に分離する。なお、本実施形態では連結画素を検知し、該連結画素の外接矩形領域の形状・サイズ・画素密度等を用いて、属性毎の領域に分割するものとするが、その他の領域分割手法を用いても構わない。

#### 【0031】

文字領域については、文字段落ごとの纏まった塊をブロックとして矩形ブロック（文字領域矩形ブロック）にセグメント化し、線画領域では、表、図形等の個々のオブジェクト（表領域矩形ブロック、線画領域矩形ブロック）ごとに矩形ブロックにセグメント化する。

#### 【0032】

ハーフトーンで表現される写真領域は、画像領域矩形ブロック、背景領域矩形ブロック等のオブジェクトごとに、矩形ブロックにセグメント化する。

#### 【0033】

これらの矩形ブロックの情報を「領域分割情報」という。

#### 【0034】

ステップS305：入力画像にオリジナルデータファイルのポインタ情報が埋め込まれているか否かの判定を行うため、OCR、OMR処理を行う。

#### 【0035】

原稿画像中に付加情報として記録された2次元バーコード、あるいはURLに該当するオブジェクトを検出し、OCRによってURLを文字認識し、OMRによって2次元バーコードを解読して入力画像のオリジナルデータファイルが格納されている記憶装置内のポインタ情報を検出する。

#### 【0036】

ポインタ情報を付加する手段は2次元バーコードに限定されるものではなく、隣接文字列間隔の変化として情報を埋め込む方法、ハーフトーンの画像に埋め込む方法等、直接視認されない、いわゆる電子透かしによる方法等がある。

**【0037】**

ステップS306（ポインタ情報抽出ステップ）：ステップS305におけるOCR、OMR、もしくは電子透かしの情報から、ポインタ情報を抽出する。

**【0038】**

ステップS307：ステップS306でポインタ情報が取得されたか否か判断する。ポインタ情報が取得されたときは、オリジナルデータファイルに直接アクセスすべく、ステップS308に分岐する。

**【0039】**

ステップS306でポインタ情報が抽出されなかったときはステップS309に進む。

**【0040】**

ステップS308：ポインタ情報が抽出されたときは、そのポインタ情報を用いてオリジナルデータファイル（電子ファイル）を検索する。オリジナルデータファイルは、電子ファイルの形態で、図1のクライアントPC102のハードディスク内、文書管理サーバ106内のデータベース105内、MFP100に設けられた記憶装置111等に格納されており、ステップS306で得られたアドレス情報（ポインタ情報）に従ってこれらの記憶装置内を検索する。検索結果としてオリジナルデータファイルがみつからなかったとき、あるいは抽出されたオリジナルデータファイルがラスターデータファイルあるいは、BMP、tiffに代表される、ラスターデータを符号化したイメージデータファイルであったときは、ステップS309に分岐する。オリジナルデータファイルを抽出できたときは、ステップS315にジャンプする。

**【0041】**

ステップS309（文書検索処理ステップ）：ポインタ情報が抽出されなかったとき、ポインタ情報に基づきオリジナルデータファイルが抽出されなかったとき、あるいは抽出されたオリジナルデータファイルがイメージデータファイルであったときは、ステップS303でマニュアル入力された検索情報あるいはステップS305のOCR処理で抽出された重要単語等に基づいて、データベース105のキーワード検索や全文検索を実行し、データベースのファイルとの類似度（

検索スコア) を求める。また、ブロックセレクション処理で抽出されたオブジェクト属性やレイアウト情報、また検索情報としてのファイルサイズ、作成日付等に関してデータベースのファイルとの類似度(検索スコア)を求める。

#### 【0042】

ステップS310：ステップS309の検索の結果は各検索条件について重み付けをした検索スコアの累積和(総合検索スコア)を算出する。所定値よりも高い類似度のファイルが複数抽出されたときは、スコアの高い順、すなわち類似度の高い順に、候補データファイルとして操作画面116においてサムネイル等で表示し、ユーザの選択を促す。ユーザが入力操作によって候補データファイルの中からオリジナルデータファイルの特定を行ったときに、データファイルを特定する。候補データファイルが1個で総合検索スコアが高い場合、ステップS310をバイパスして自動的にステップS311にジャンプするようにしてもよい。

#### 【0043】

ステップS311：ステップS309あるいはステップS310で1個のオリジナルデータファイルが特定されたか否か判断する。1個のデータファイルが特定されたときはステップS315にジャンプし、データファイルが抽出されなかったとき、あるいは抽出されたデータファイルがイメージデータファイルのときはステップS312に進む。

#### 【0044】

ステップS312(ベクトル化ステップ)：ベクトル化処理により各特定領域のイメージデータをベクトルデータに変換する。

#### 【0045】

ベクトル化の方法は、以下(a)～(f)等が存在する。

(a) 特定領域が文字領域のとき、さらにOCRによる文字画像のコード変換を行ったり、あるいは文字のサイズ、スタイル、字体を認識し、原稿を走査して得られた文字に可視的に忠実なフォントデータに変換する。

(b) 特定領域が文字領域であり、かつOCRによる認識が不可能であったとき、文字の輪郭を追跡し、輪郭情報(アウトライン)を線分のつながりとして表現する形式に変換する。

(c) 特定領域が図形領域のとき、図形オブジェクトの輪郭を追跡し、輪郭情報を線分のつながりとして表現する形式に変換する。

(d) b、cの線分形式のアウトライン情報をベジエ関数などでフィッティングして、関数情報に変換する。

(e) cの図形オブジェクトの輪郭情報から、図形の形状を認識し、円、矩形、多角形、等の図形定義情報に変換する。

(f) 特定領域が図形領域であって、特定領域の表形式のオブジェクトの場合、罫線や枠線を認識し、所定のフォーマットの帳票フォーマット情報に変換する。

#### 【0046】

以上の方針以外に、ラスターデータを所定のコマンドやコード情報に置きえる種々のベクトル化処理が考えられる。

#### 【0047】

ステップS313：ステップS310のベクトルデータをそのまま使用して、アプリケーションデータ形式に変換し、出力する。通常、データ形式は使用するアプリケーションに依存し、目的に応じたファイル形式に変換する必要がある。

#### 【0048】

編集等再利用可能なアプリケーションデータ形式とは、ワードプロセッサ、表計算ソフトウェア等のアプリケーションソフトウェアであり、例えば、マイクロソフト・コーポレーションのワードプロセッサWORD（登録商標）や表計算アプリケーションソフトウェアEXCEL（登録商標）などがある。これらのアプリケーションはそれぞれ使用する目的が異なっており、目的に応じたファイル形式が定義され、その形式でファイル（データ）を保存する。

#### 【0049】

より汎用的なファイル形式としては、マイクロソフト・コーポレーションのRTF（リッチ・テキスト・ファイル）形式、近年使用されるようになったSVG（Scalable Vector Graphics）形式、あるいは単純にテキストデータのみを扱うプレーンテキスト形式、などが知られており、これらは対応するアプリケーションにおいて共通に使用可能である。

#### 【0050】

ステップ S 314：ステップ S 313 で生成されたベクトル化された領域のベクトルデータ、およびその他の領域のイメージデータを電子ファイルとして記憶装置 111 に格納する。イメージデータは例えば J P E G 等の形式で格納される。

#### 【0051】

ステップ S 315：データの格納場所を示すアドレスを出力する。ステップ S 314 でデータを記憶装置 111 に格納したときは、該記憶装置 111 に格納したデータのアドレスが出力され、ステップ S 308 または S 311 でオリジナルデータファイルが抽出されたときはオリジナルデータファイルのアドレスを出力する。

#### 【0052】

ステップ S 316：データベース 105、記憶装置 111 のデータに関してはあらかじめインデックスファイルが生成されており、以上の処理において格納場所が判明し、あるいは新たな格納場所に格納されたデータについて、これらの格納場所をインデックスファイルに追加する。

#### 【0053】

さらに、インデックスファイルには、ステップ S 303 で入力された単語や、ステップ S 313 で自動抽出された単語が登録される。これによって、次回、検索を行う際の検索性能が向上する。

#### 【0054】

インデックスファイルにおいて、ステップ S 308、S 311 でオリジナルデータファイルが抽出されたときにも、ステップ S 303 で入力された単語は追加登録される。これによって、ポインタ情報を使用できないような状況でも検索性能が向上する。

#### 【0055】

ステップ S 317：検出されあるいは生成された電子ファイルの内容について印刷等の記録出力であるかユーザに確認し、記録出力であるときはステップ S 318 に進み、記録出力以外の出力であるときはステップ S 320 にジャンプする。

**【0056】**

ステップS318：電子ファイル等にポインタ情報を付加する。ポインタ情報付加は、2次元バーコードにより出力画像に付加する方法、文字列中やハーフトーン画像中に電子透かしを埋め込む方法等、種々の公知方法により実行可能である。

**【0057】**

これによって印刷された画像が読み込まれたとき、直ちにポインタ情報を取得でき、オリジナルデータファイルにアクセスし得る。

**【0058】**

ステップS319：ステップS318のポインタ情報に加えて、電子ファイルに関連した情報を出力画像に付加する。情報付加の方法はステップS318と同様である。これによって、ポインタ情報が使用できない状況においても効率的にオリジナルデータファイルを検索し得る。

**【0059】**

ステップS320：以上の処理により得られた電子ファイルを用いて、文書の加工、蓄積、伝送、記録等の種々の処理を行う。生成され、あるいは取得された電子ファイルは、イメージデータファイルに比較してデータサイズがコンパクトであり、蓄積効率向上、伝送時間短縮、表示・記録品質向上の効果が得られる。

**【0060】**

次に図3の主要なステップについて詳細に説明する。

**【0061】****[ブロックセレクションステップ]**

ステップS302（ブロックセレクションステップ）においては、図4右半部の画像42に示すように、入力画像を属性ごとに矩形ブロックに分割する。前述のように、矩形ブロックの属性としては、文字(TEXT)/図面(PICTURE)/線画(Line)/表(Table)/写真(Photo)等がある。

**【0062】**

ブロックセレクションステップにおいては、まず入力画像を白黒に2値化し、黒画素輪郭で囲まれる画素塊を抽出する。

**【0063】**

さらに、このように抽出された黒画素塊の大きさを評価し、大きさが所定値以上の黒画素塊の内部にある白画素塊に対する輪郭追跡を行う。白画素塊に対する大きさ評価、内部黒画素塊の追跡というように、内部の画素塊が所定値以上である限り、再帰的に内部画素塊の抽出、輪郭追跡を行う。

**【0064】**

画素塊の大きさは、例えば画素塊の面積によって評価される。

**【0065】**

このようにして得られた画素塊に外接する矩形ブロックを生成し、矩形ブロックの大きさ、形状に基づき属性を判定する。

**【0066】**

例えば、縦横比が1に近く、大きさが一定の範囲の矩形ブロックは文字領域矩形ブロックの可能性がある文字相当ブロックとし、近接する文字相当ブロックが規則正しく整列しているときに、これら文字相当ブロックを纏めた新たな矩形ブロックを生成し、新たな矩形ブロックを文字領域矩形ブロックとする。

**【0067】**

また扁平な画素塊を線画領域矩形ブロック、一定大きさ以上でかつ四角形の白画素塊を整列よく内包する黒画素塊を表領域矩形ブロック、不定形の画素塊が散在している領域を写真領域矩形ブロック、それ以外の不定形の画素塊を画像領域矩形ブロックとする。

**【0068】**

ブロックセレクションステップでは、このようにして生成された矩形ブロックのそれぞれについて、図5に示す、属性等のブロック情報および入力ファイル情報を生成する。

**【0069】**

図5において、ブロック情報には各ブロックの属性、位置の座標X、座標Y、幅W、高さH、OCR情報が含まれる。属性は1～5の数値で与えられ、1は文字領域矩形ブロック、2は図面領域矩形ブロック、3は表領域矩形ブロック、4は線画領域矩形ブロック、5は写真領域矩形ブロックを示す。座標X、座標Yは

入力画像における各矩形ブロックの始点のX、Y座標（左上角の座標）である。幅W、高さHは矩形ブロックのX座標方向の幅、Y座標方向の高さである。OCR情報は入力画像におけるポインタ情報の有無を示す。

#### 【0070】

さらに入力ファイル情報として矩形ブロックの個数を示すブロック総数Nが含まれる。

#### 【0071】

これらの矩形ブロックごとのブロック情報は、特定領域でのベクトル化に利用される。またブロック情報によって、特定領域とその他の領域を合成する際の相対位置関係を特定でき、入力画像のレイアウトを損なわずにベクトル化領域とラスターデータ領域を合成することが可能となる。

#### 【0072】

##### [ポインタ情報抽出ステップ]

ステップS307（ポインタ情報抽出ステップ）は図6の各ステップによって実行される。図7は、ポインタ情報抽出ステップの処理対象となる原稿310を示し、原稿310はデータ処理装置115内のページメモリ（図示せず。）に格納される。原稿311には、文字領域矩形ブロック312、313、画像領域矩形ブロック314、2次元バーコード（QRコード）のシンボル311が記録されている。

#### 【0073】

ステップS701：まず、記憶装置111のページメモリに格納された原稿310の入力画像をCPU（図示せず。）で走査して、ブロックセレクションステップの処理結果から、2次元バーコードシンボル311の位置を検出する。

#### 【0074】

QRコードシンボルは、4隅のうちの3隅に、特定の位置検出要素パターンが設けられ、位置検出要素パターンを検出することにより、QRコードシンボルを検出し得る。

#### 【0075】

ステップS702：次に、位置検出パターンに隣接する形式情報を復元し、シ

シンボルに適用されている誤り訂正レベルおよびマスクパターンを得る。

#### 【0076】

ステップS703：次にシンボルの型番（モデル）を決定する。

#### 【0077】

ステップS704：ステップS702の形式情報から得られたマスクパターンを使って、QRコードのシンボルの符号化領域ビットパターンをXOR演算し、QRコードのシンボルのマスク処理を解除する。

#### 【0078】

ステップS705：ステップS703で得られたモデルに基づいて配置規則を取得し、この配置規則に基づいて、シンボルキャラクタを読み取り、メッセージのデータ及び誤り訂正コード語を復元する。

#### 【0079】

ステップS706：復元されたメッセージについて、誤り訂正コード語に基づいて、誤りがあるかどうかの検出を行う。誤りが検出されたときは、訂正のためにステップS707に分岐する。

#### 【0080】

ステップS707：復元されたメッセージを訂正する。

#### 【0081】

ステップS708：誤り訂正されたデータより、モード指示子および文字数指示子に基づいて、データコード語をセグメントに分割し、データコード語を復元する。

#### 【0082】

ステップS709：検出された仕様モードに基づいてデータコード文字を復号し、結果を出力する。

#### 【0083】

2次元バーコードに組み込まれたデータは、オリジナルデータファイルのポインタ情報を表しており、例えばファイルサーバー名およびファイル名からなるパス情報で構成される。あるいは、対応するファイルへのURL等や、ファイルが格納されたデータベース105内あるいは記憶装置111内のファイルID等

で構成される。

#### 【0084】

本実施形態ではポインタ情報を2次元バーコードの形で付与した原稿310について説明したが、種々のポインタ情報の記録形態を採用し得る。

#### 【0085】

例えば、所定のルールに従った文字列によってポインタ情報を直接文字列で記録し、ブロックセレクションステップによって文字列の矩形ブロックを検出してもよい。検出された文字列を認識することによりポインタ情報を取得し得る。

#### 【0086】

あるいは文字領域矩形ブロック312や文字領域矩形ブロック313において、透かし情報として隣接文字列の間隔に視認し難い程度の変調を加えて印刷出力されている文書をスキャンして得たイメージデータから、該文字列間隔変調の情報によってポインタ情報を表現し得る。このような透かし情報は、後述する文字認識処理を行う際に各文字の間隔を検出することによって検出でき、ポインタ情報を取得し得る。また写真領域矩形ブロック314の中に電子透かしとしてポインタ情報を附加する事も可能である。

#### 【0087】

##### [ポインタ情報による電子ファイル検索]

図3のステップ308、S311における、ポインタ情報による電子ファイル検索は図8の各ステップにより実行される。

#### 【0088】

ステップS901：ポインタ情報に含まれるアドレスに基づいて、ファイルサーバを特定する。クライアントPC102、データベース105、文書管理サーバ106、記憶装置111を内蔵するMFP100のうちの少なくともいずれかがファイルサーバとして使用される。アドレスは、URLや、サーバ名およびファイル名よりなるパス情報である。

#### 【0089】

ステップS902：ステップS901によって特定されたファイルサーバに対して、アドレスを転送する。

**【0090】**

ステップS903：ステップS901で特定されたファイルサーバは、ステップS902で転送されたアドレスを受信し、このアドレスに基づいてオリジナルデータファイルを検索する。

**【0091】**

ステップS904：ステップS903のファイル検索によってオリジナルデータファイルを抽出できたか否か判断する。ファイルを抽出できたときはステップS905に進み、ファイルを抽出できなかったときはステップS906に進む。

**【0092】**

ステップS905：図3に関連して説明したように、ファイルのアドレスをMFP100に通知し、ユーザがオリジナルファイルデータの取得を希望するときはオリジナルデータファイルをMFP100に転送する。これによって処理を終了する。

**【0093】**

ステップS906：ステップS903でファイルを抽出できなかったときはその旨をMFP100に通知し、処理を終了する。

**【0094】****[ファイル検索処理]**

図3のステップS309のファイル検索処理は、入力画像にポインタ情報が含まれていなかった場合、またはポインタ情報から電子ファイルを抽出できなかつた場合に行われる。ファイル検索処理は、キーワード検索処理、全文検索処理、レイアウト検索処理、条件絞り込み検索処理等、複数の検索方法の組み合わせにより実行される。

**【0095】**

キーワード検索処理は、検索対象のイメージデータの検索キーを用いて、データベース内の各電子ファイルに関連したキーワードがあらかじめ登録されているインデックスファイルを検索（一致検索あるいは曖昧一致検索）する処理である。

**【0096】**

なお、検索キーには、ステップS303でユーザによりマニュアル入力された単語と、ステップS305のOCR処理で入力画像から抽出した文字中の単語、電子透かしに埋め込まれた単語等が用いられる。

#### 【0097】

全文検索処理は、キーワード検索処理の検索キーによって、オリジナルデータファイルのテキスト情報全文に対し検索を行う処理であり、抽出された検索キーの個数が多いほど類似度が高いと判定される。

#### 【0098】

条件絞り込み検索は、ステップS303でマニュアル入力されたサイズ情報、日付情報等の条件に基づいて電子ファイルを絞り込む検索処理である。

#### 【0099】

レイアウト検索処理については、図10を参照しつつ詳細に説明する。

#### 【0100】

[レイアウト検索処理]

レイアウト検索処理は、図10の各ステップによって実行される。

#### 【0101】

ここでは、ステップS302の結果、抽出された各矩形ブロック及び入力画像データが、図5に示すブロック情報および入力ファイル情報を備えるものとする。

#### 【0102】

ブロック情報では、例えば、矩形ブロックは座標Xの小さい順に配列（X座標が同じブロックについては更にY座標の小さい順で配列）され、ブロック1、ブロック2、ブロック3、ブロック4、ブロック5、ブロック6の座標Xは、 $X_1 \leq X_2 \leq X_3 \leq X_4 \leq X_5 \leq X_6$  の大小関係になっている。これらの情報を使用して、データベース内から、入力画像に類似したファイルのレイアウト検索の処理は、図10の各ステップによって実行される。ここで、データベースファイルは、図5と同様の情報を備えることを前提とする。フローチャートの流れは、ブロック情報および入力ファイル情報と、データベース中のファイルとを順次比較するものである。

**【0103】**

ステップS1101：後述する類似率などの初期化を行う。

**【0104】**

ステップS1102：次に、入力画像におけるブロック総数Nとの差が所定値以内、すなわちデータベース内のデータファイルにおいて、 $N - \Delta N < n < N + \Delta N$ なるn個のブロック総数を持つデータファイルであるか判断する。条件に合致するデータファイルが検索抽出されたときにはステップS1103に移行して当該検索抽出されたデータファイル内の矩形ブロックの情報と入力画像内のブロックとを順次比較していく。一方、ブロック総数の差が大きいときはステップS1114にジャンプする。矩形ブロックの情報比較においては、ステップS1103～S1109によって、属性類似率、サイズ類似率、OCR類似率をそれぞれ算出し、これら類似率に基づいてステップS1111において、該比較対象のデータファイルの総合類似率を算出する。

**【0105】**

ステップS1103：入力画像のブロック属性と、ステップS1102で抽出されたデータベースファイルのブロック属性を比較する。ブロック属性が一致したときはステップS1104に進み、ブロック属性が一致しなかったときは、ステップS1110に進み、入力画像の全ブロックについての比較が終了したか判断する。

**【0106】**

ステップS1104：ステップS1103で比較された、ブロック情報の属性について属性類似率を更新する。属性類似率の算出方法は、任意の方法を用いることが可能であるが、例えば、 $|(\text{属性一致ブロック数}) / (\text{総ブロック数})|$ に基づいて算出するものとする。

**【0107】**

ステップS1105：入力画像のブロック情報のサイズ（幅W、高さH）と、ステップS1102で抽出されたデータベースファイルのサイズ（幅w、高さh）を比較する。サイズの差が所定範囲、すなわち $W - \Delta W < w < W + \Delta W$ でありかつ $H - \Delta H < h < H + \Delta H$ であるか否か判断する。所定範囲内のときはステッ

PS1106に進み、所定範囲内でなかったときはステップS1110に進み、  
入力画像の全ブロックについての比較が終了したか判断する。

#### 【0108】

ステップS1106：ステップS1105で比較されたブロック情報のサイズについてサイズ類似率を更新する。サイズ類似率の算出方法は、任意の方法を用いることが可能であるが、例えば、 $\{1 - (\text{サイズ差}) / (\text{入力イメージデータの該ブロックサイズ})\}$ で各ブロックのサイズ類似率を求めて、更に、該データファイルについてサイズ類似度の平均値を求めることにより算出するものとする。

#### 【0109】

ステップS1107：入力画像のブロック情報、およびステップS1102で抽出されたデータファイルのブロック情報にOCR情報が「有」か否かを判断する。両者がOCR情報「有」であったときはステップS1108に進み、OCR情報「無」のときはステップS1110に進み、入力画像の全ブロックについての比較が終了したか判断する。

#### 【0110】

ステップS1108：入力画像のブロック情報のOCR情報と、ステップS1102で抽出されたデータファイルのOCR情報を比較する。

#### 【0111】

ステップS1109：ステップS1108で比較されたOCR情報についてOCR情報類似率を算出する。OCR情報類似率の算出方法は、任意の方法を用いることが可能であるが、例えば、各ブロックの認識結果文字同士を比較し、認識結果の一致率を求めるなどして求めることとする。

#### 【0112】

ステップS1110：入力画像に含まれる全てのブロックについて、当該データファイルの各ブロックとの比較処理が終了したか否かを判断する。全ての矩形ブロックの処理が終了したときはステップS1111に進み、未処理の矩形ブロックが残っているときはステップS1115を経て、次のブロックの比較処理を行うためにステップS1103に戻る。

**【0113】**

ステップS1111：属性類似率、サイズ類似率、OCR類似率に基づいて総合類似率を算出する。総合類似率の算出方法は、例えば、各類似率に所定の重み付けを行って、和をとることにより算出される。

**【0114】**

ステップS1112：総合類似率が予め設定された閾値Thより高いか否か判断する。総合類似率が閾値Thより高ければステップS1113に進み、総合類似率が閾値Thより高くないときはステップS1114に進む。

**【0115】**

ステップS1113：ステップS1112で総合類似率が閾値より高いと判断されたデータベースのデータファイルを類似候補として保存する。

**【0116】**

ステップS1114：データベースの全てのデータファイルについて処理が終了したか否か判断する。全てのデータベースファイルの処理が終了したときはそのまま処理を終了し、未処理のデータベースファイルが残っているときはステップS1116を経てステップS1101に戻る。

**【0117】**

ステップS1115：データファイルの個々のブロック情報と、入力画像のブロック情報の全てを比較するように、比較対象のブロックを次のブロックに進める。

**【0118】**

ステップS1116：データベースの各データファイルが順次比較されるよう、次のデータファイルを比較対象にする。

**【0119】**

図10の各ステップにおいて、N、W、Hは、入力画像のブロック総数、各ブロック幅、各ブロック高さであり、 $\Delta N$ 、 $\Delta W$ 、 $\Delta H$ は、入力画像のブロック情報を基準とした許容範囲示す。 $n$ 、 $w$ 、 $h$ は、データベースファイルのブロック総数、各ブロック幅、各ブロック高さである。

**【0120】**

なお、不図示ではあるが、ステップS1105にてサイズ比較時に、座標X、座標Yの位置情報の比較などを行ってもよい。

#### 【0121】

以上の複数の検索方式による検索結果は、各々類似度スコアとして数値化され、最終的な検索結果は各スコアに重み付けした合計値を、総合検索類似度として評価する。重み付けの方法は、ユーザがステップS303で入力した情報に基づく検索結果を重くし、あるいは過去の検索結果の統計処理によって有意とされたスコアを重くし、あるいは重み入力用のユーザインターフェースを設け、ユーザが任意に重み設定する方法であってもよい。

#### 【0122】

総合検索類似度は所定の閾値と比較され、閾値よりも高いスコアの電子ファイルが検索対象候補とされる。複数候補が抽出されたときは、ステップS310において候補選択の処理を行う。

#### 【0123】

##### [ベクトル化ステップ]

ステップS312（ベクトル化ステップ）では、ファイルサーバにオリジナルデータファイルが存在しないときに、図4のイメージデータ41を、各矩形ブロックごとにベクトル化する。矩形ブロックが文字領域矩形ブロックであるときには、ベクトル化された各文字について文字認識処理を実行する。

#### 【0124】

ベクトル化処理は図9の各ステップにより実行される。

#### 【0125】

ステップS1001：特定領域が文字領域矩形ブロックであるか否か判断し、文字領域矩形ブロックであればステップS1002以下のステップに進み、パターンマッチングの一手法を用いて認識を行い、対応する文字コードを得る。特定領域が文字領域矩形ブロックでないときは、ステップS1012の処理に移行する。

#### 【0126】

ステップS1002：特定領域に対し横書き、縦書きの判定（組み方向判定）

をおこなうために、特定領域内で画素値に対する水平・垂直の射影を取る。

#### 【0127】

ステップS1003：ステップS1002の射影の分散を評価する。水平射影の分散が大きい場合は横書き、垂直射影の分散が大きい場合は縦書きと判断する。

#### 【0128】

ステップS1004：ステップS1003の評価結果に基づき、組み方向を判定し、行の切り出しを行い、その後文字を切り出して文字画像を得る。

#### 【0129】

文字列および文字への分解は、横書きならば水平方向の射影を利用して行を切り出し、切り出された行に対する垂直方向の射影から、文字を切り出す。縦書きの文字領域に対しては、水平と垂直について逆の処理を行う。行、文字切り出しに際して、文字のサイズも検出し得る。

#### 【0130】

ステップS1005：ステップS1004で切り出された各文字について、文字画像から得られる特徴を数十次元の数値列に変換した観測特徴ベクトルを生成する。特徴ベクトルの抽出には種々の公知手法があり、例えば、文字をメッシュ状に分割し、各メッシュ内の文字線を方向別に線素としてカウントしたメッシュ数次元ベクトルを特徴ベクトルとする方法がある。

#### 【0131】

ステップS1006：ステップS1005で得られた観測特徴ベクトルと、あらかじめ字種ごとに求められている辞書特徴ベクトルとを比較し、観測特徴ベクトルと辞書特徴ベクトルとの距離を算出する。

#### 【0132】

ステップS1007：ステップS1006で算出された距離を評価し、最も距離の近い字種を認識結果とする。

#### 【0133】

ステップS1008：ステップS1007における距離評価において、最短距離が所定値よりも大きいか否か判断する。距離が所定値以上の場合は、辞書特徴

ベクトルにおいて、形状が類似する他の文字に誤認識している可能性が高い。そこで距離が所定値以上の場合は、ステップS1007の認識結果を採用せず、ステップS1011の処置に進む。距離が所定値より小のときは、ステップS1007の認識結果を採用し、ステップ1009に進む。

#### 【0134】

ステップS1009（フォント認識ステップ）：文字認識の際に用いる、字種数ぶんの辞書特徴ベクトルを、文字形状種すなわちフォント種に対して複数用意しておき、パターンマッチングの際に、文字コードとともにフォント種を出力することで、文字フォントを認識し得る。

#### 【0135】

ステップS1010：文字認識およびフォント認識によって得られた文字コードおよびフォント情報を用いて、各々あらかじめ用意されたアウトラインデータを用いて、各文字をベクトルデータに変換する。なお、入力画像がカラーの場合は、カラー画像から各文字の色を抽出してベクトルデータとともに記録する。

#### 【0136】

ステップS1011：文字を一般的な線画と同様に扱い、該文字をアウトライン化する。すなわち誤認識を起こす可能性の高い文字については、可視的にイメージデータに忠実なアウトラインのベクトルデータを生成する。

#### 【0137】

ステップS1112：特定領域が文字領域矩形ブロックでないときは、画像の輪郭に基づいてベクトル化の処理を実行する。

#### 【0138】

以上の処理により、文字領域矩形ブロックに属するイメージ情報をほぼ形状、大きさ、色が忠実なベクトルデータに変換出来る。

#### 【0139】

##### [文字領域以外の領域のベクトル化]

ステップS1012の文字領域矩形ブロック以外の領域、すなわち図面領域矩形ブロック、線画領域矩形ブロック、表領域矩形ブロック等と判断されたときは、特定領域内で抽出された黒画素塊の輪郭をベクトルデータに変換する。

**【0140】**

文字領域以外の領域のベクトル化においては、まず線画等を直線および／または曲線の組み合わせとして表現するために、曲線を複数の区間（画素列）に区切る「角」を検出する。角とは曲率が極大となる点であり、図11の曲線上の画素 $P_i$ が角か否かの判断は以下のように行う。

**【0141】**

すなわち、 $P_i$ を起点とし、曲線に沿って $P_i$ から両方向に所定画素（ $k$ 個とする。）ずつ離れた画素 $P_{i-k}$ 、 $P_{i+k}$ を線分 $L$ で結ぶ。画素 $P_{i-k}$ 、 $P_{i+k}$ 間の距離を $d_1$ 、線分 $L$ と画素 $P_i$ との距離を $d_2$ 、曲線の画素 $P_{i-k}$ 、 $P_{i+k}$ 間の弧の長さを $A$ とするとき、 $d_2$ が極大となるとき、あるいは比（ $d_1/A$ ）が閾値以下となるときに画素 $P_i$ を角と判断する。

**【0142】**

角によって分割された画素列を、直線あるいは曲線で近似する。直線への近似は最小二乗法等により実行し、曲線への近似は3次スプライン関数などを用いる。画素列を分割する角の画素は近似直線あるいは近似曲線における、始端または終端となる。

**【0143】**

さらにベクトル化された輪郭内に白画素塊の内輪郭が存在するか否かを判断し、内輪郭が存在するときはその輪郭をベクトル化し、内輪郭の内輪郭というよう再帰的に反転画素の内輪郭をベクトル化する。

**【0144】**

以上のように、輪郭の区分線近似を用いれば、任意形状の図形のアウトラインをベクトル化することができる。元原稿がカラーの場合は、カラー画像から図形の色を抽出してベクトルデータとともに記録する。

**【0145】**

図12に示すように、ある注目区間で外輪郭 $P_R j$ と、内輪郭 $P_R j + 1$ あるいは別の外輪郭が近接している場合、2個あるいは複数の輪郭線をひとまとめにし、太さを持った線として表現することができる。例えば、輪郭 $P j + 1$ の各画素 $P_i$ から輪郭 $P_R j$ 上で最短距離となる画素 $Q_i$ までの距離 $P_i Q_i$ を算出し

、 $P_i Q_i$  のばらつきがわずかである場合には、注目区間を画素 $P_i$ 、 $Q_i$ の中点 $M_i$ の点列に沿った直線または曲線で近似し得る。近似直線、近似曲線の太さは、例えば距離 $P_i Q_i$ の平均値とする。

#### 【0146】

線や線の集合体である表罫線は、太さを持つ線の集合とすることにより、効率よくベクトル表現することができる。

#### 【0147】

輪郭まとめの処理の後、全体の処理を終了する。

#### 【0148】

なお写真領域矩形ブロックについては、ベクトル化せず、イメージデータのまとめる。

#### 【0149】

##### [図形認識]

以上の線図形等のアウトラインをベクトル化した後、ベクトル化された区分線を図形オブジェクトごとにグループ化する。

#### 【0150】

図13の各ステップは、ベクトルデータを図形オブジェクトごとにグループ化する処理を示す。

#### 【0151】

ステップS1501：まず各ベクトルデータの始点、終点を算出する。

#### 【0152】

ステップS1502（図形要素検出）：ステップS1501で求められた始点、終点情報を用いて、図形要素を検出する。図形要素とは、区分線が構成している閉図形であり、検出に際しては、始点、終端近傍において連結ベクトルの端点を捜す。すなわち、閉形状を構成する各ベクトルはその両端にそれぞれ連結するベクトルを有しているという原理を応用する。

#### 【0153】

ステップS1503：次に図形要素内に存在する他の図形要素、もしくは区分線をグループ化し、一つの図形オブジェクトとする。また、図形要素内に他の図

形要素、区分線が存在しない場合は図形要素を図形オブジェクトとする。

#### 【0154】

##### [図形要素の検出]

ステップS1502（図形要素検出）の処理は、図14の各ステップによって実行される。

#### 【0155】

ステップS1601：まず、ベクトルデータより両端に連結していない不要なベクトルを除去し、閉図形を構成するベクトルを抽出する。

#### 【0156】

ステップS1602：次に閉図形を構成するベクトルについて、いずれかのベクトルの端点（始点または終点）を開始点とし、一定方向、例えば時計回りに、順にベクトルを探索する。すなわち、他端点において他のベクトルの端点を探索し、所定距離内の最近接端点を連結ベクトルの端点とする。閉図形を構成するベクトルを1まわりして開始点に戻ったとき、通過したベクトルを全て一つの図形要素を構成する閉図形としてグループ化する。また、閉図形内部にある閉図形構成ベクトルも全てグループ化する。さらにまだグループ化されていないベクトルの始点を開始点とし、同様の処理を繰り返す。

#### 【0157】

ステップS1603：最後に、ステップS1601で除去された不要ベクトルのうち、ステップS1602で閉図形としてグループ化されたベクトルに端点が近接しているベクトルを検出し、一つの図形要素としてグループ化する。

#### 【0158】

以上の処理によって図形ブロックを、再利用可能な個別の図形オブジェクトとして扱う事が可能になる。

#### 【0159】

以上のベクトル化処理は、通常、入力画像全体に対して行う必要性は低く、ユーザが指定した特定領域のみをベクトル化だけに対して行えば充分な場合が多い。

#### 【0160】

ユーザが指定する特定領域のみに対してベクトル化処理を行うことにより、処理のパフォーマンスを向上し、かつユーザが所望する部分のみを効率よくベクトル化して、次のステップの検索処理に利用したり、もしくはイメージ情報のうちの必要な部分だけの再編集／再利用をする効果を提供することが可能になる。

#### 【0161】

[アプリデータへの変換処理]

図3のブロックセレクションステップ（ステップS304）の後、ベクトル化（ステップS312）した結果のデータを用いて、ステップS313のアプリデータへの変換処理を実行する。ステップS312のベクトル化処理結果は図15に示す中間データの形式、いわゆるドキュメント・アナリシス・アウトプット・フォーマット（DAOF）と呼ばれる形式で保存されている。

#### 【0162】

図15において、DAOFは、ヘッダ1701、レイアウト記述データ部1702、文字認識記述データ部1703、表記述データ部1704、画像記述データ部1705よりなる。

#### 【0163】

ヘッダ1701には、処理対象の入力画像に関する情報が保持される。

#### 【0164】

レイアウト記述データ部1702には、入力画像中の矩形ブロックの属性TEX（文字）、TITLE（タイトル）、CAPTION（キャプション）、LINE（線画）、PICTURE（画像）、FRAME（枠）、TABLE（表）、PHOTO（写真）等の情報と、これら属性が認識された各矩形ブロックの位置情報が保持される。

#### 【0165】

文字認識記述データ部1703には、TEXT、TITLE、CAPTION等の文字領域矩形ブロックを文字認識して得られる文字認識結果が保持される。

#### 【0166】

表記述データ部1704には、表領域矩形ブロックTABLEの表構造の詳細が格納される。

**【0167】**

画像記述データ部1705には、図面領域矩形ブロックPICTUREや線画領域矩形ブロックLINE等のブロックにおけるイメージデータが、入力画像データから切り出して保持される。

**【0168】**

このようなDAOFは中間データとしてのみならず、それ自体ファイル化され保存されることもあり、このファイル状態では、一般的な文書作成アプリケーションソフトウェアのオブジェクトにおける再利用は不可能である。そこで、DAOFをアプリケーションデータに変換する。

**【0169】**

アプリデータへの変換処理は図16の各ステップにより実行される。

**【0170】**

ステップS1801：DAOF形式のデータを入力する。

**【0171】**

ステップS1802：アプリデータの元となる文書構造ツリー生成を行う。

**【0172】**

ステップS1803：文書構造ツリーを元に、DAOF内の実データを取得し、実際のアプリデータを生成する。

**【0173】**

ステップS1803の文書構造ツリー生成処理は図17の各ステップにより実行される。図17の処理における全体制御の基本ルールとして、処理の流れはミクロブロック（単一矩形ブロック）からマクロブロック（矩形ブロックの集合体）へ移行する。以後「矩形ブロック」は、ミクロブロックおよびマクロブロック両者を意味するものとする。

**【0174】**

ステップS1901：矩形ブロック単位で、縦方向の関連性に基づいて、矩形ブロックを再グループ化する。図17の処理は繰り返し実行されることがあるが、処理開始直後はミクロブロック単位での判定となる。

**【0175】**

ここで、関連性とは、距離が近い、ブロック幅（横方向の場合は高さ）がほぼ同一であることなどの特徴によって定義される。また、距離、幅、高さなどの情報はD A O F を参照し、抽出する。

#### 【0176】

図18の入力画像では、最上部で、矩形ブロックT1、T2が横方向に並列されている。矩形ブロックT1、T2の下には横方向セパレータS1が存在し、横方向セパレータS1の下に矩形ブロックT3、T4、T5、T6、T7が存在する。

#### 【0177】

矩形ブロックT3、T4、T5は、横方向セパレータS1下側の領域における左半部において上から下に、縦方向に配列され、矩形ブロックT6、T7は、横方向セパレータS1下側の領域における右半部において上下に配列されている。

#### 【0178】

ステップS1901の縦方向の関連性に基づくグルーピングの処理を実行する。これによって、矩形ブロックT3、T4、T5が1個のグループ（矩形ブロック）V1にまとめられ、矩形ブロックT6、T7が1個のグループ（矩形ブロック）V2にまとめられる。グループV1、V2は同一階層となる。

#### 【0179】

ステップS1902：縦方向のセパレータの有無をチェックする。セパレータは、D A O F 中でライン属性を持つオブジェクトであり、アプリケーションソフトウェア中で明示的にブロックを分割する機能をもつ。セパレータを検出すると、処理対象の階層において、入力画像の領域を、セパレータを境界として左右に分割する。図18では縦方向のセパレータは存在しない。

#### 【0180】

ステップS1903：縦方向のグループ高さの合計が入力画像の高さに等しくなったか否か判断する。すなわち縦方向（例えば上から下へ。）に処理対象の領域を移動しながら、横方向のグルーピングを行うとき、入力画像全体の処理が終了したときには、グループ高さ合計が入力画像高さになることを利用し、処理の終了判断を行う。グルーピングが終了したときはそのまま処理終了し、グルーピ

ングが終了していなかったときはステップS1904に進む。

#### 【0181】

ステップS1904：横方向の関連性に基づくグルーピングの処理を実行する。これによって、矩形ブロックT1、T2が1個のグループ（矩形ブロック）H1にまとめられ、矩形ブロックV1、V2が1個のグループ（矩形ブロック）H2にまとめられる。グループH1、H2は同一階層となる。ここでも、処理開始直後はミクロブロック単位での判定となる。

#### 【0182】

ステップS1905：横方向のセパレータの有無をチェックする。セパレータを検出すると、処理対象の階層において、入力画像の領域を、セパレータを境界として上下に分割する。図18では横方向のセパレータS1が存在する。

#### 【0183】

以上の処理結果は図19のツリーとして登録される。

#### 【0184】

図19において、入力画像V0は、最上位階層にグループH1、H2、セパレータS1を有し、グループH1には第2階層の矩形ブロックT1、T2が属する。

#### 【0185】

グループH2には、第2階層のグループV1、V2が属し、グループV1には、第3階層の矩形ブロックT3、T4、T5が属し、グループV2には、第3階層の矩形ブロックT6、T7が属する。

#### 【0186】

ステップS1906：横方向のグループ長合計が入力画像の幅に等しくなったか否か判断する。これによって横方向のグルーピングに関する終了判断を行う。横方向のグループ長がページ幅となっている場合は、文書構造ツリー生成の処理を終了する。横方向のグループ長がページ幅となっていないときは、ステップS1901に戻り、再びもう一段上の階層で、縦方向の関連性チェックから繰り返す。

#### 【0187】

図18、図19のツリー構造の段階で、横方向の分割幅がページ幅になっているので、そのまま処理を終了し、最後にページ全体を表す最上位階層のV0を文書構造ツリーに付加する。

#### 【0188】

文書構造ツリーが完成した後、その情報を元にステップS1803においてアプリデータの生成を行う。

#### 【0189】

図18、図19に基づくアプリデータによる、アプリケーションソフトウェアの処理の例は以下のとおりである。

#### 【0190】

まず、グループH1が横方向に2個の矩形ブロックT1、T2を有するので、2カラムとし、T1のDAOFを参照し、その内部情報（文字認識結果の文章、画像等）を出力する。その後、カラムを変え、T2の内部情報を出力し、続いてセパレータS1を出力する。

#### 【0191】

次グループH2の処理に移る。グループH2は横方向に2個の矩形ブロックV1、V2を有するので、2カラムとして出力する。グループV1については、矩形ブロックT3、T4、T5の順にその内部情報を出力し、その後カラムを変え、グループV2の矩形ブロックT6、T7の内部情報を出力する。

#### 【0192】

以上によりアプリデータへの変換処理が実行される。

#### 【0193】

##### [ポインタ情報の付加]

ステップS318のポインタ情報付加の処理は、抽出されあるいは生成されたファイルを記録する際に、記録紙にポインタ情報を付加する処理である。ポインタ情報を参照することによって、電子ファイルを容易に抽出し得る。

#### 【0194】

図20のフローチャートにおいて、ポインタ情報を2次元バーコード（JIS X0510に基づくQRコードシンボル等）を付加する処理を説明する。

**【0195】**

図7に関連して説明したように、2次元バーコードは、対応する電子ファイルを取得し得る場所を示す情報よりなる。

**【0196】**

ステップS2201：QRコードシンボルに変換すべきポインタ情報を示す文字を識別すべき、ポインタ情報を分析するとともに、誤り検出及び誤り訂正レベルを設定し、ポインタ情報を収容し得る最小型番を選択する。

**【0197】**

ステップS2202：ステップS2201で分析したポインタ情報を所定ビット列に変換し、必要に応じて、ポインタ情報のモード（数字、英数字、8ビットバイト、漢字等）を示す指示子、および終端パターンを付加する。さらに、このように得られたデータをビットコード語に変換する。

**【0198】**

ステップS2203：ステップS2202で生成したビットコード語列を、型番および誤り訂正レベルに応じて、所定数のブロックに分割し、各ブロックごとに誤り訂正コード語を生成する。さらに誤り訂正コード語をビットコード語列の後付加する。

**【0199】**

ステップS2204：ステップS2203で生成した各ブロックのビットコード語を接続し、各ブロックに誤り訂正コード語を付加する。さらに必要に応じて、各ブロックの訂正コードの後に剩余コード語を付加する。以上によりコード語モジュールが生成される。

**【0200】**

ステップS2205：所定のマトリクスに、位置検出パターン、分離パターン、タイミングパターン、位置合わせパターン、およびコード語モジュールを配列する。

**【0201】**

ステップS2206：ステップS2205で生成されたマトリクスにおけるシンボル符号化領域に対して、最適なマスクパターンを選択し、マトリクスとマス

クパターンのXOR演算によってモジュールを生成する。

#### 【0202】

ステップS2207：ステップS2206で生成されたモジュールに対して、形式情報、型番情報を生成し、QRコードシンボルを完成する。

#### 【0203】

このようなアドレス情報が組み込まれたQRコードシンボルは、クライアントPC102からの要求により、電子ファイルをMFP112で印刷する際に、データ処理装置115内で記録可能なラスタデータに変換され、印刷画像における所定位置に画像形成される。

#### 【0204】

ステップS306に関連して説明したように、画像形成された印刷画像を画像読み取り装置110で読み取ることにより、ポインタ情報が取得でき、電子ファイルの格納場所を検出し得る。

#### 【0205】

なお以上の実施形態においては、画像読み取り装置110等から入力された入力画像の処理について説明したが、記憶媒体に格納されたラスターデータあるいはその符号化データよりなるイメージデータ、通信手段によって供給されたイメージデータ等、入力画像以外のイメージデータについても、本発明は有効である。

#### 【0206】

本発明に係る画像処理方法を実施する手段は図1、図2に記載した画像処理システムに限定されるものではなく、専用の画像処理装置、汎用コンピュータ等種々の手段を採用し得る。

#### 【0207】

汎用コンピュータにおいて実行する際には、画像処理方法の各ステップを汎用コンピュータに実行させるためプログラムコードを含むコンピュータ実行可能なプログラムを、汎用コンピュータに読み込ませる。

#### 【0208】

汎用コンピュータが画像処理を実行するためのプログラムは、汎用コンピュー

タに内蔵されたROMや、汎用コンピュータが読み取り得る記憶媒体から読み込まれ、あるいはネットワークを通じてサーバ等から読み込まれる。

### 【0209】

また本発明の趣旨と範囲は、本発明の特定の説明と図に限定されるものではなく、本願特許請求の範囲に述べられた内容の様々な修正および変更に及ぶことは当業者にとって理解されるであろう。

### 【0210】

本発明の実施態様の例を以下に示す。

### 【0211】

[実施態様1] 入力画像に対応するオリジナルデータファイルを検索するための画像処理方法であって、ユーザにより入力された情報に基づいて前記入力画像に関する第1の検索情報を取得する第1検索情報取得ステップと、前記入力画像に含まれる特徴データを第2の検索情報として取得する第2検索情報取得ステップと、前記第1の検索情報と前記第2の検索情報とを用いて、前記入力画像に対応するオリジナルデータファイルを検索する検索ステップと、を有することを特徴とする画像処理方法。

### 【0212】

[実施態様2] 更に、前記第1の検索情報を、前記オリジナルデータファイルを検索するためのインデックスとしてインデックスファイルに登録するインデックス登録ステップを有することを特徴とする実施態様1に記載の画像処理方法。

### 【0213】

[実施態様3] 前記第1の検索情報は、検索のためのキーワードであることを特徴とする実施態様1に記載の画像処理方法。

### 【0214】

[実施態様4] 前記第1の検索情報は、前記オリジナルデータファイルのデータサイズであることを特徴とする実施態様1に記載の画像処理方法。

### 【0215】

[実施態様5] 前記第1の検索情報は、前記オリジナルデータファイルの日

付情報であることを特徴とする実施態様 1 に記載の画像処理方法。

#### 【0216】

〔実施態様 6〕 前記第 2 の検索情報は、前記入力画像内のポインタ情報に基づいて抽出される前記オリジナルデータファイルの格納場所に関する情報であることを特徴とする実施態様 1 に記載の画像処理方法。

#### 【0217】

〔実施態様 7〕 前記第 2 の検索情報は、前記入力画像内の文字領域について文字認識処理を行うことによって得た、文字認識結果の文字コードであることを特徴とする実施態様 1 に記載の画像処理方法。

#### 【0218】

〔実施態様 8〕 前記第 2 の検索情報は、前記入力画像を領域分割して得た各ブロックの特徴データであることを特徴とする実施態様 1 に記載の画像処理方法。

#### 【0219】

〔実施態様 9〕 更に、前記検索ステップでオリジナルデータファイルが検索できなかった場合、前記入力画像をベクトルデータに変換するベクトル化ステップを有することを特徴とする実施態様 1 に記載の画像処理方法。

#### 【0220】

〔実施態様 10〕 前記ベクトルデータに変換された入力画像を、アプリケーションソフトウェアで取り扱い得る形式に変換するステップをさらに備えることを特徴とする実施態様 9 に記載の画像処理方法。

#### 【0221】

〔実施態様 11〕 前記ベクトルデータに変換された入力画像をデータベースに格納する格納ステップを更に有することを特徴とする実施態様 9 に記載の画像処理方法。

#### 【0222】

〔実施態様 12〕 前記第 1 の検索情報を、前記格納ステップでデータベースに格納されたベクトルデータの画像を検索するためのインデックスとして、インデックスファイルに登録するインデックス登録ステップを有することを特徴とする実施態様 10 に記載の画像処理方法。

**【0223】**

〔実施態様13〕 更に、前記検索されたオリジナルデータファイルを出力する出力ステップを有し、前記出力ステップでは、当該出力されるオリジナルデータファイルにポインタ情報を附加して出力することを特徴とする実施態様1に記載の画像処理方法。

**【0224】**

〔実施態様14〕 前記ポインタ情報を電子透かしとして前記オリジナルデータファイルに附加することを特徴とする実施態様13記載の画像処理方法。

**【0225】**

〔実施態様15〕 前記検索ステップでは、キーワード検索、全文検索、レイアウト検索の少なくともいずれかを用いて、前記オリジナルデータファイルを検索することを特徴とする実施態様1記載の画像処理方法。

**【0226】**

〔実施態様16〕 入力画像に対応するオリジナルデータファイルを検索するための画像処理システムであって、ユーザにより入力された情報に基づいて前記入力画像に関する第1の検索情報を取得する第1検索情報取得手段と、前記入力画像に含まれる特徴データを第2の検索情報として取得する第2検索情報取得手段と、前記第1の検索情報と前記第2の検索情報とを用いて、前記入力画像に対応するオリジナルデータファイルを検索する検索手段と、を有することを特徴とする画像処理システム。

**【0227】**

〔実施態様17〕 実施態様1乃至15のいずれかに記載の画像処理方法の各ステップを汎用コンピュータに実行させるためのプログラムコードを含むコンピュータ実行可能なプログラム。

**【0228】**

〔実施態様18〕 実施態様17記載のコンピュータ実行可能なプログラムが格納されたコンピュータ読み取り可能な記憶媒体。

**【0229】****【発明の効果】**

本発明によれば、紙文書に対応するオリジナルデータファイルを検索する際の検索性能が向上する。

【図面の簡単な説明】

【図1】 本発明に係る画像処理方法の第1実施形態を実施する画像処理システムを示すブロック図である。

【図2】 図1におけるMFPを示すブロック図である。

【図3】 本発明に係る画像処理方法の一実施形態を示すフローチャートである。

【図4】 図3の画像処理方法により処理される原稿と処理結果の画像を示す図である。

【図5】 ブロックセレクション処理で得られるブロック情報および入力ファイル情報を示す表である。

【図6】 原稿の画像からポインタ情報を抽出する処理を示すフローチャートである。

【図7】 ポインタ情報を含む原稿の画像を示す図である。

【図8】 図3のポインタ情報に基づくファイル検索処理を示すフローチャートである。

【図9】 図3の文字領域のベクトル化の処理を示すフローチャートである。

。

【図10】 図3のファイル検索の処理を示すフローチャートである。

【図11】 図9のベクトル化の処理における角抽出の処理を示す図である。

。

【図12】 図9のベクトル化の処理における輪郭線まとめの処理を示す図である。

【図13】 図9のベクトル化で生成されたベクトルデータのグループ化の処理を示すフローチャートである。

【図14】 図13の処理でグループ化されたベクトルデータに対する図形要素検出の処理を示すフローチャートである。

【図15】 ベクトル化処理結果のデータを示すマップである。

【図16】 図3のアプリデータ変換の処理を示すフローチャートである。

【図17】 図16の文書構造ツリー生成の処理を示すフローチャートである。

【図18】 文書構造ツリー生成処理の対象となる文書を示す図である。

【図19】 図17の処理によって生成される文書構造ツリーを示す図である。

【図20】 図3のポインタ情報付加の処理を示すフローチャートである。

【符号の説明】

100 MF P

105 データベース

106 文書管理サーバ

110 画像読み取り装置

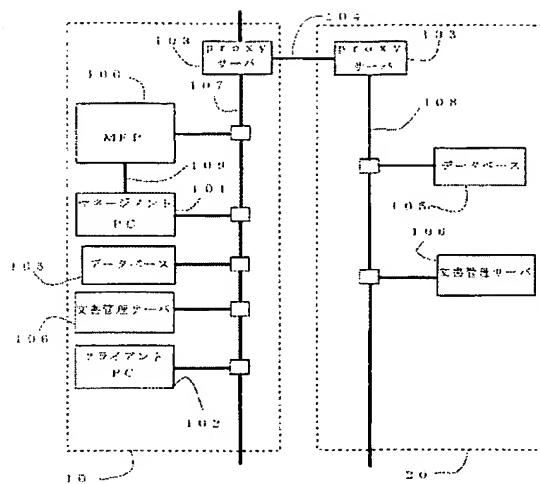
111 記憶装置

112 記憶装置

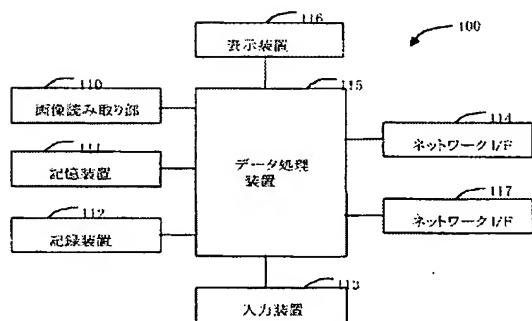
115 データ処理装置

## 【書類名】 図面

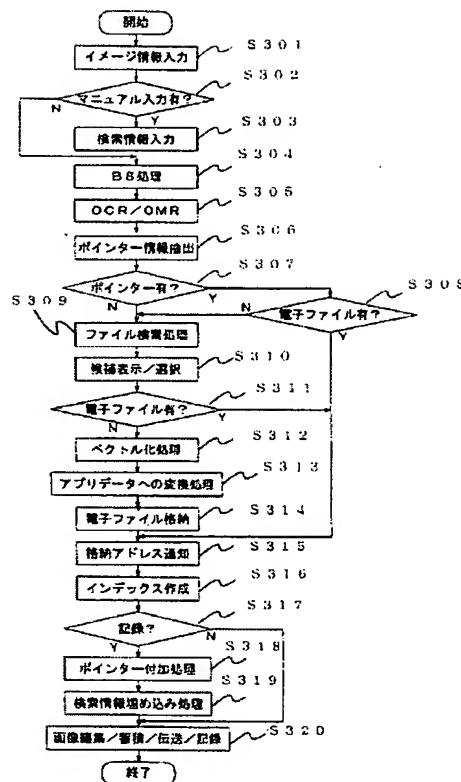
【図 1】



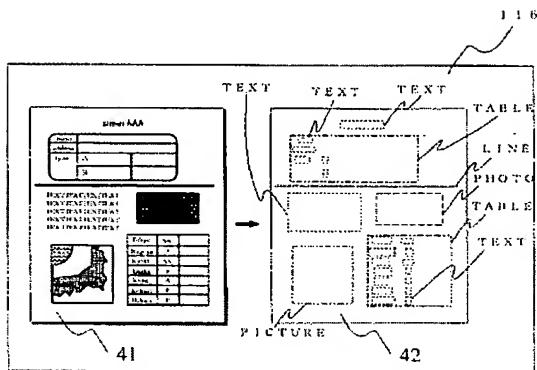
【図 2】



【図 3】



【図 4】



【図 5】

ブロック情報

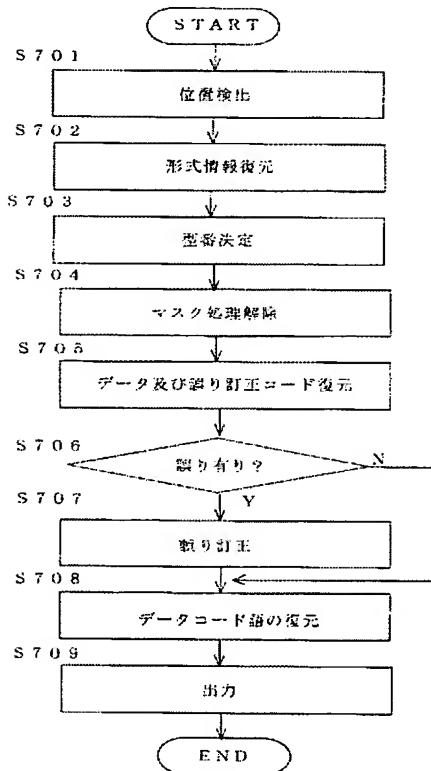
	属性	座標X	座標Y	幅W	高さH	OCR情報
ブロック 1	1	X 1	Y 1	W 1	H 1	有
ブロック 2	3	X 2	Y 2	W 2	H 2	有
ブロック 3	2	X 3	Y 3	W 3	H 3	無
ブロック 4	1	X 4	Y 4	W 4	H 4	有
ブロック 5	3	X 5	Y 5	W 5	H 5	有
ブロック 6	5	X 6	Y 6	W 6	H 6	無

属性: 1=text, 2=picture, 3=table, 4=line, 5=photo

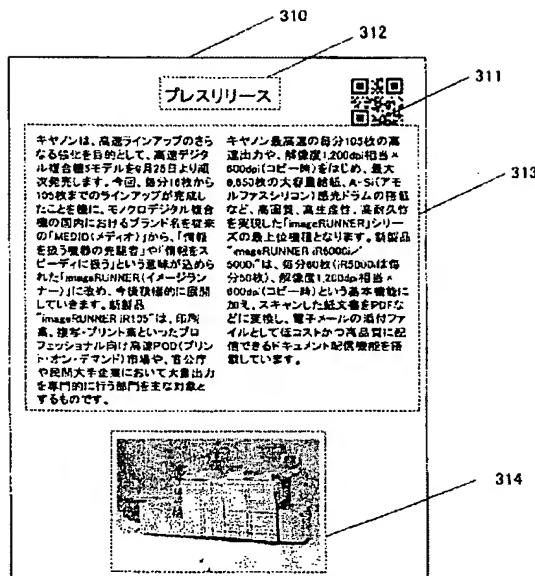
入力ファイル情報

ブロック記数	N (= 6)

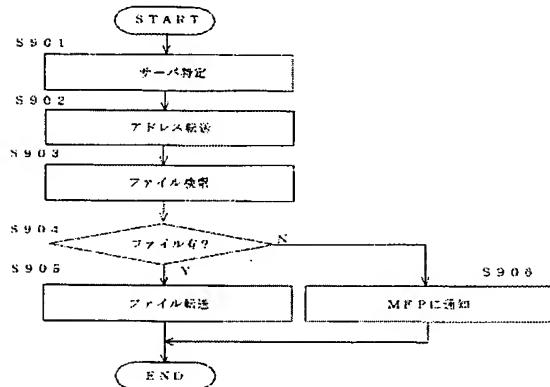
【図 6】



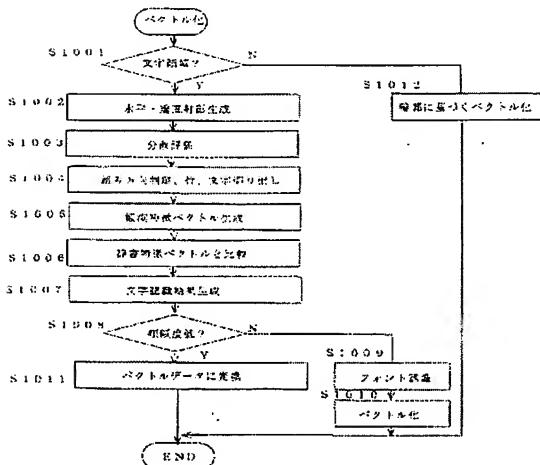
〔四七〕



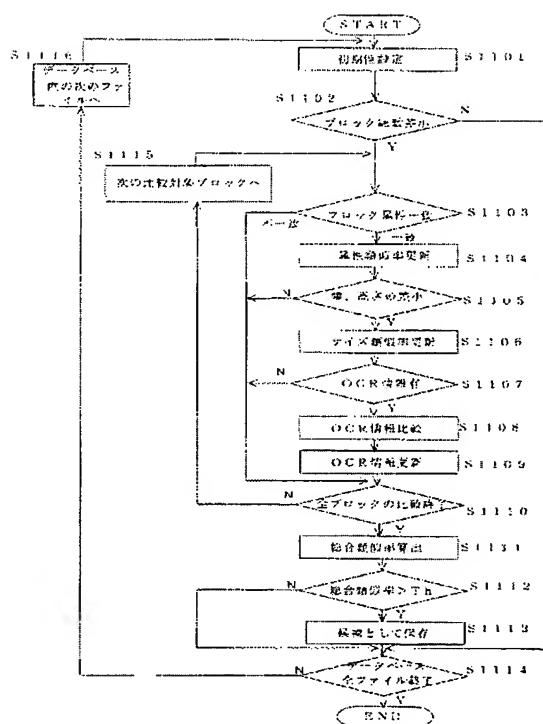
(义 8)



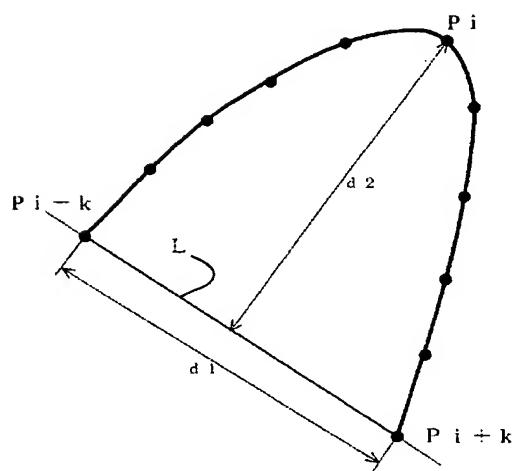
〔四九〕



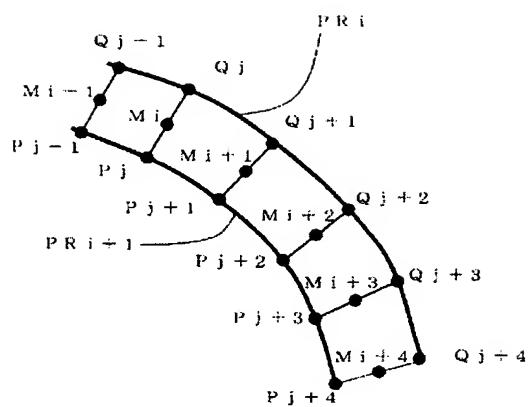
【図10】



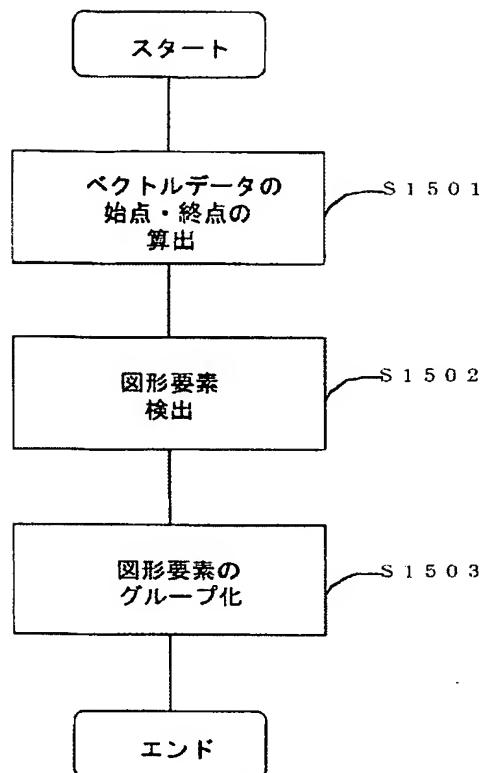
【図11】



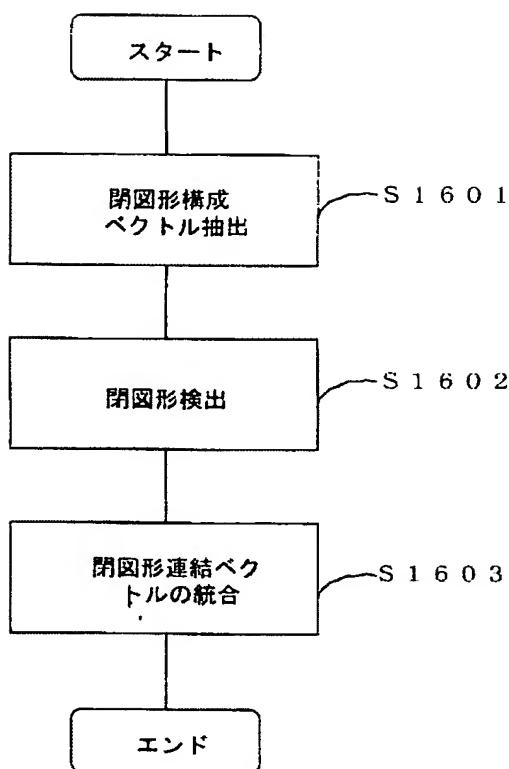
【図12】



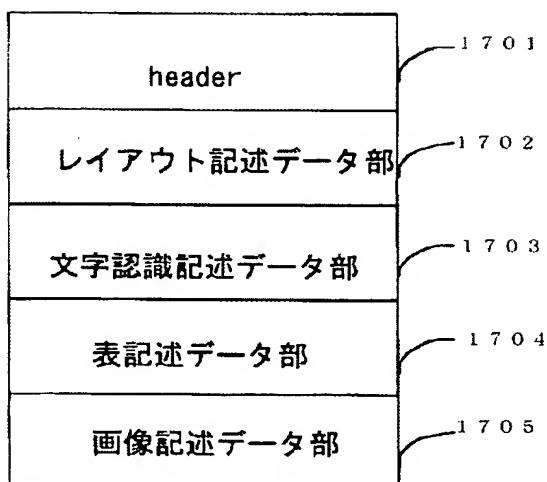
【図13】



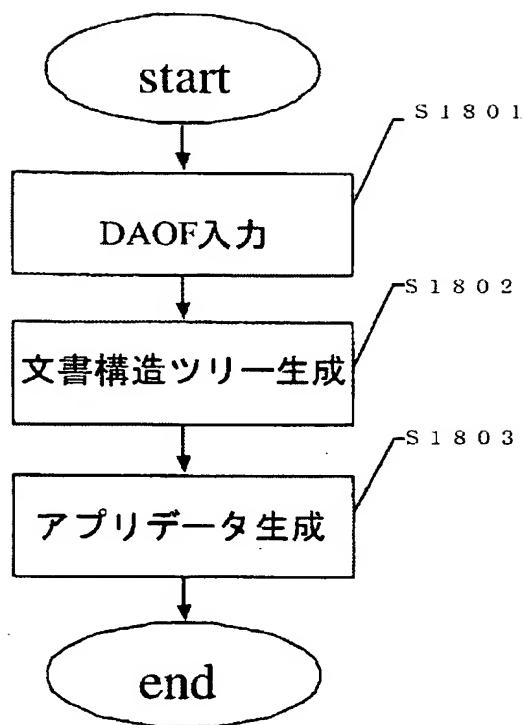
【図14】



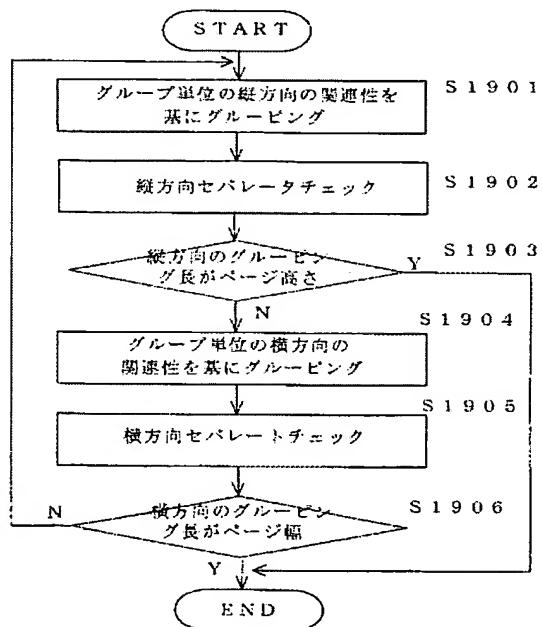
【図15】



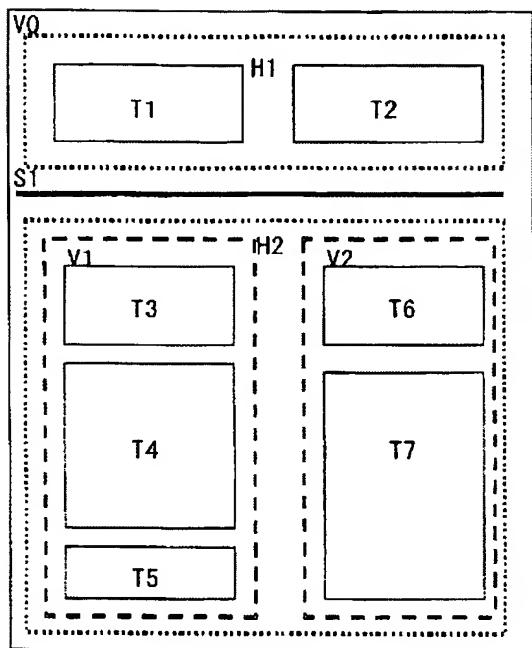
【図16】



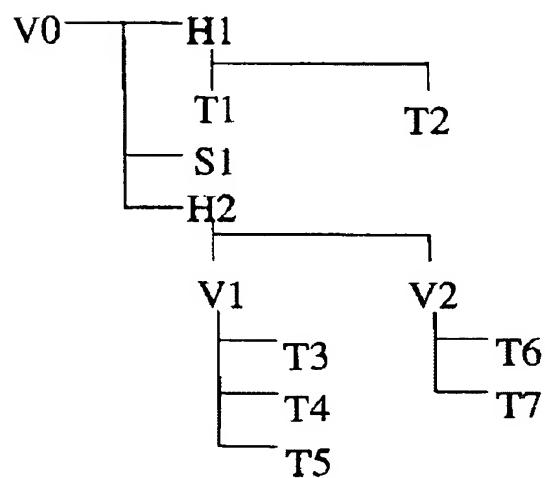
【図17】



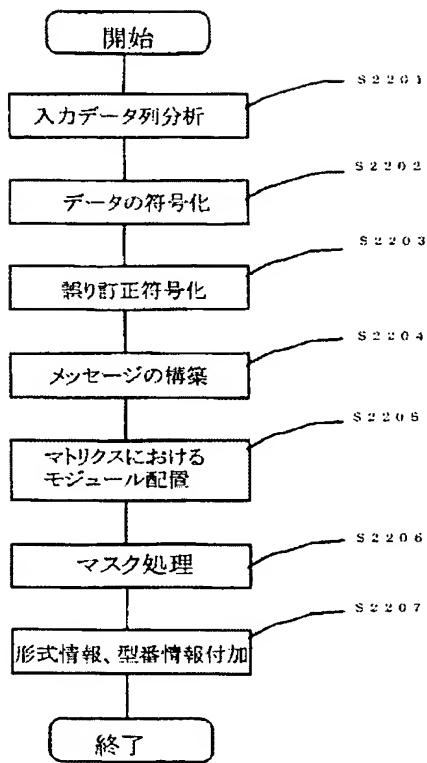
【図18】



【図19】



【図20】



【書類名】 要約書

【要約】

【課題】 紙文書の画像等種々の形態のイメージデータを、検索および再利用が容易な電子ファイルとして取り扱い可能とする。

【解決手段】 入力画像について、入力画像特定のための情報をユーザに入力させる。

【選択図】 図3

## 認定・付加情報

特許出願の番号	特願 2003-121076
受付番号	50300696211
書類名	特許願
担当官	第七担当上席 0096
作成日	平成 15 年 4 月 30 日

## &lt;認定情報・付加情報&gt;

## 【特許出願人】

【識別番号】 000001007

【住所又は居所】 東京都大田区下丸子3丁目30番2号

【氏名又は名称】 キヤノン株式会社

## 【代理人】

【識別番号】 100079832

【住所又は居所】 東京都千代田区鍛冶町1-6-15 共同ビル（  
神田駅前）22号 つくし特許事務所

【氏名又は名称】 山本 誠

## 【選任した代理人】

【識別番号】 100078846

【住所又は居所】 東京都千代田区鍛冶町1丁目6番15号 共同ビル  
(神田駅前) 22号 つくし特許事務所

【氏名又は名称】 大音 康毅

## 【選任した代理人】

【識別番号】 100087583

【住所又は居所】 東京都千代田区神田鍛冶町3-3-9 共同ビル  
(新千代田) 73号 つくし特許事務所

【氏名又は名称】 田中 増顯

次頁無

特願 2003-121076

出願人履歴情報

識別番号 [000001007]

1. 変更年月日 1990年 8月30日

[変更理由] 新規登録

住 所 東京都大田区下丸子3丁目30番2号  
氏 名 キヤノン株式会社